ASSIGNMENT 4

Handout: **Tuesday, 8 November 2016**
Due: **11:30 am, Thursday, 17 November 2016**

GOALS:

- To understand better inheritance, polymorphism, and dynamic binding;
- To design and implement Java classes;
- To get used to the IntelliJ Idea IDE;

1. CALCULATOR LIBRARY

The following code demonstrates how a user would like to use two classes `Literal` and `BinaryExpr` to construct and evaluate binary expressions:

```
Literal l1 = new Literal(1);
Literal l2 = new Literal(3);
Literal l3 = new Literal(5);
System.out.println(l2);                  // print out: 3.0
System.out.println(l2.evaluate());       // print out: 3.0
BinaryExpr b1 = new BinaryExpr(l2, l1, BinaryOp.DIVIDE);
System.out.println(b1);                  // print out: 3.0 / 1.0
BinaryExpr b2 = new BinaryExpr(b1, l3, BinaryOp.ADD);
System.out.println(b2);                  // print out: (3.0 / 1.0) + 5.0
System.out.println(b2.evaluate());       // print out: 8.0
BinaryExpr b3 = new BinaryExpr(b1, b2, BinaryOp.MULTIPLY);
System.out.println(b3.evaluate());       // print out: 24.0
```

Enum `BinaryOp` defines four constants `DIVIDE`, `MULTIPLY`, `ADD`, and `MINUS`.

WHAT TO DO:

Task 1: Please design and implement the two classes and other supporting code to meet the user's requirements, i.e., to support the operations used in the code snippet above. (10 points)

Task 2: Suppose later on the user wants to extend the existing code to also support the usage as shown below:

```
Variable v1 = new Variable("a");         // Each variable has a single character as its name
Expr b4 = new BinaryExpr(b2, v1, BinaryOp.DIVIDE);
System.out.println(b4);                  // print out: ((3.0 / 1.0) + 5.0) / a
Environment env = new Environment();
env.addVariable("a", 2.0);               // variable "a" has value 2.0 in env
System.out.println(b4.evaluate());       // throws RuntimeException
System.out.println(b4.evaluate(env));    // print out: 4.0
```

Explain what changes you need to apply to your design and implementation from Task 1 to support the new requirements. *Note you do NOT have to implement the changes.* (10 points)

Task 3 (Bonus): Implement the changes as you have described in **Task 2** to actually support the new requirements. (10 points)

WHAT TO HAND IN:

Task 1: Your IntelliJ IDEA project named “Task1” containing all the code to support the requirements in this task;

Task 2: A text file named “Task2.txt” describing the necessary changes;

Task 3 (Bonus): Your IntelliJ IDEA project named “Task3” containing all the code to support requirements in Task 1 and 3. If you hand in project “Task3”, you don’t have to hand in “Task1”.

Zip all your solutions into an archive Assignment4.zip, and submit this archive on Blackboard.